BEA WebLogic Adapter for SWIFT

Installation and Configuration Guide for WebLogic Integration 2.1

Release 7.0
Document Date: October 2002

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA WebLogic Adapter for SWIFT Installation and Configuration Guide for WebLogic Integration 2.1

Part Number	Date
N/A	October 2002

Table of Contents

About This Document

1. Installing the Adapter for WebLogic Integration 2.1

About This Document

This document explains how to install the BEA WebLogic Adapter for SWIFT for WebLogic Integration 2.1, which is used to develop client-server and server-to-server, n-tier interfaces between SWIFT standardized systems. It describes how to install the BEA WebLogic Adapter for SWIFT using the MQSeries transport within WebLogic Integration interfaces between SWIFT and other Web applications.

This document is organized as follows:

- Chapter 1, "Installing the Adapter for WebLogic Integration 2.1," describes how to install the BEA WebLogic Adapter for SWIFT.

Audience

This document is written for system integrators who develop client interfaces between SWIFT message systems and other Web applications. It describes how to install the BEA WebLogic Adapter for SWIFT to develop application environments with a specific focus on message integration. It is assumed that readers know Web technologies and have a general understanding of SWIFT messages and Microsoft Windows and UNIX systems.

Related Information

The BEA corporate Web site provides all documentation for WebLogic Server and WebLogic Integration. For information about these products, go to `http://e-docs.bea.com`. Documents that you may find helpful when installing the BEA WebLogic Adapter for SWIFT are:

- *BEA WebLogic Adapter for SWIFT User Guide*
- *BEA WebLogic Adapter for SWIFT Release Notes*
- BEA WebLogic Server 6.1 installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/wls/docs61/index.html`
- BEA WebLogic Integration 2.1 installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/wlintegration/v2_1sp/index.html`
- *BEA Application Explorer Installation Guide*

Contact Us!

Your feedback on the BEA WebLogic Integration documentation is important to us. Send us e-mail at `docsupport@bea.com` if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Integration documentation.

In your e-mail message, please indicate which version of the BEA WebLogic Adapter for SWIFT documentation you are using.

If you have any questions about this version of BEA WebLogic Integration, or if you have problems installing and running BEA WebLogic Integration, contact BEA Customer Support through BEA WebSupport at `www.bea.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

Convention	Item
monospace boldface text	Identifies significant words in code. *Example*: `void **commit** ( )`
monospace italic text	Identifies variables in code. *Example*: `String *expr*`
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f *file-list*]... [-l *file-list*]...`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
. . .	Indicates one of the following in a command line: ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f *file-list*]... [-l *file-list*]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

1 Installing the Adapter for WebLogic Integration 2.1

You can use the BEA WebLogic Adapter for SWIFT to integrate SWIFT messages and SWIFT systems into your WebLogic Integration Java 2 Enterprise Edition (J2EE) application server environment. The installation in this manual is based on a transport protocol of IBM WebSphere MQ (that is, MQSeries). For help with other transports, see the related adapter installation document. Installation is dependant on the underlying resources of WebLogic Server and WebLogic Integration, as well as on IBM MQ MQSeries. After this infrastructure is in place, installation includes the addition of a J2EE Enterprise Application Archive (`.ear` file), and the creation of an Application View (AppView) for the event adapter and the service adapter.

This section includes the following topics:

Before Installing the Adapter

The BEA WebLogic Adapter for SWIFT supports many different transport protocols for SWIFT formatted messages. The transport protocol of choice must be installed prior to installation of the BEA WebLogic Adapter for SWIFT. This document will describe the process for one sample transport, IBM's MQSeries. For other transports, please see the related documentation for the associated transport.

Before you install the BEA WebLogic Adapter for SWIFT, you must review the *BEA WebLogic Adapter for* SWIFT *Release Notes* to ensure that you have the required prerequisite software installed. The *BEA WebLogic Adapter for* SWIFT *Release Notes* is available at the following URL:

`http://edocs.bea.com/wladapters/doc70/index.html`

Understanding the Representation of Paths

Because the location of files in the WebLogic Integration environment depends on options selected during installation and configuration, the conventions that follow are used throughout to represent paths.

- *`BEA_HOME`* represents the BEA Home directory specified for your WebLogic installation.

  For example, if you install the product in the default location on a Windows system, *`BEA_HOME`* represents `c:\bea`.

- *WLI_HOME* represents the root of your WebLogic Integration installation.

  For example:

  - If you install WebLogic Integration 2.1 in the default location on a Windows system, *WLI_HOME* represents `c:\bea\wlintegration2.1`.

- *domain* is used to indicate the name of a domain.

  - In WebLogic Integration 2.1, preconfigured domains (`bpmdomain`, `eaidomain`, `wlidomain`, and `samples`) are created as subdirectories of the *WLI_HOME*`\config` directory. Therefore, *domain* may be used to represent the root of a preconfigured WebLogic Integration 2.1 domain as follows:

    WLI_HOME`\config\`*domain*

- *DOMAIN_HOME* represents the complete path to the root of a domain.

  For example:

  - If you install WebLogic Integration 2.1 in the default location on a Windows system, *DOMAIN_HOME* represents `c:\bea\wlintegration2.1\config\`*domain*.

Note: *WLI_HOME* and *BEA_HOME* (italicized) also represent the corresponding Windows and UNIX environment variables. For example, the literal interpretation of *WLI_HOME* is `%WLI_HOME%` for Windows and `$WLI_HOME` for UNIX.

Unlike *WLI_HOME* and *BEA_HOME*, *DOMAIN_HOME* is not an environment variable that is set by default in the WebLogic Integration environment.

Step 1. Obtaining the BEA WebLogic Adapter for SWIFT

The BEA WebLogic Adapter for SWIFT is packaged as an EAR file (`BEA_SWIFT_1_0.ear`). You can obtain the software on CD or download it from `www.bea.com`.

Obtain the BEA WebLogic Adapter for SWIFT `BEA_SWIFT_1_0.ear` file by downloading it or mounting the installation media on the WebLogic Server Administration Console client machine (that is, the machine on which a user will configure and manage WebLogic Integration, typically an administrator's PC running a Web browser).

Note: At the time of publication of this document, WebLogic Server 6.1 is unable to explode RAR files (BEA Support CASE number 333672). Use an extraction tool (such as WinZip) to extract the contents of the Adapter EAR file, `BEA_SWIFT_1_0.ear` and add the location of the unpackaged objects to the server's class path (see Step 2. Extracting JARs and Adjusting the Classpath).

Step 2. Extracting JARs and Adjusting the Classpath

Set the classpath using the procedure appropriate for your system:

- Extracting JARs and Adjusting the Classpath for Windows
- Extracting JARs and Adjusting the Classpath for UNIX

Extracting JARs and Adjusting the Classpath for Windows

To unzip the adapter JAR files and adjust the classpath on Windows, complete the following steps:

1. Use WinZip (or another similar extracting product) to extract the `BEA_SWIFT_1_0.ear` file to the directory of your choice (for example, *`BEA_HOME`*`\ear\lib\`*`SWIFT`*).
2. Go to the root directory for your domain:

   `cd` *`DOMAIN_HOME`*

   Note: You must choose a WebLogic Integration domain that supports application integration functionality.
3. Find the `Set`*`DomainType`*`Data.cmd` file.

   Here, *`DomainType`* is the type of the domain. For example, if your domain is configured to support the development of solutions that employ the full range of WebLogic Integration functionality, it contains the `SetwliDomainData.cmd` file.
4. Update the following `SVRCP` environment variable settings to the `Set`*`DomainType`*`Data.cmd` file for the domain to include all the JAR files included in the EAR file.

The `SVRCP` environment variable is used in the `Set`*`DomainType`*`Data` script to set the classpath for the java executable.

After the following line:

```
set SVRCP=%SVRCP%;%WLI_DOMAIN_HOME%\wlai
```

add the following JAR files that are listed in the order required for the class path:

```
REM ===== General Adapter Java Libraries =====
set SVRCP=%SVRCP%;d:\adapters\ibi-edaqm.jar
set SVRCP=%SVRCP%;d:\adapters\xercesImpl.jar
set SVRCP=%SVRCP%;d:\adapters\xmlParserAPIs.jar
set SVRCP=%SVRCP%;d:\adapters\jdom.jar
set SVRCP=%SVRCP%;d:\adapters\dtdparser.jar
set SVRCP=%SVRCP%;d:\adapters\engine.jar
```

```
REM ===== Libraries for MQSeries =====
set SVRCP=%SVRCP%;d:\bea\ear\lib\SWIFT\BEA_SWIFT_1_0.jar
set SVRCP=%SVRCP%;"D:\Program
Files\MQSeries\Java\lib\com.ibm.mq.iiop.jar"
set SVRCP=%SVRCP%;"D:\Program
Files\MQSeries\Java\lib\com.ibm.mq.jar"
set SVRCP=%SVRCP%;"D:\Program
Files\MQSeries\Java\lib\com.ibm.mqbind.jar"
REM ===== Native Libraries and Localized Properties =====
set SVRCP=%SVRCP%;"D:\Program Files\MQSeries\Java\lib"
```

here *D:\bea\ear\lib\SWIFT* is the directory specified in step 1.

Extracting JARs and Adjusting the Classpath for UNIX

To extract the adapter JAR files and adjust the classpath on UNIX, complete the following steps:

1. Use jar (or another similar extracting product) to extract `BEA_SWIFT_1_0.ear` to a directory of your choice (for example, *BEA_HOME*`/ear/lib/`*SWIFT*).
2. Go to the root directory for your domain:

   `cd` *DOMAIN_HOME*

   Note: You must choose a WebLogic Integration domain that supports application integration functionality.
3. Find the `Set`*DomainType*`Data.cmd` file.

   Here, *DomainType* is the type of domain. For example, if your domain is configured to support the development of solutions that employ the full range of WebLogic Integration functionality, it contains the `SetwliDomainData.cmd` file.
4. Update the following `SVRCP` environment variable settings to the `Set`*DomainType*`Data.cmd` file for the domain to include all the JAR files included in the EAR file.

Note: The `SVRCP` environment variable is used in the `Set`*DomainType*`Data` script to set the classpath for the java executable.

After the following line:

```
SVRCP=$SVRCP:$WLI_DOMAIN_HOME/wlai
```

add the following JAR files, which are listed in the order required for the class path:

```
# Set SWIFT Adapter classpath

SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/ibi-edaqm.jar
SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/BEA_SWIFT_1_0.jar
SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/xercesImpl.jar
SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/xmlParserAPIs.jar
SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/engine.jar
SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/jdom.jar
SVRCP=$SVRCP:/usr/bea/ear/lib/SWIFT/dtdparser.jar
SVRCP=$SVRCP:/opt/MQSeries/Java/lib/com.ibm.mq.iiop.jar
SVRCP=$SVRCP:/opt/MQSeries/Java/lib/com.ibm.mq.jar
SVRCP=$SVRCP:/opt/MQSeries/Java/lib/com.ibm.mqbind.jar
SVRCP=$SVRCP:/opt/MQSeries/Java/lib"
```

here */usr/bea/ear/lib/SWIFT* is the directory specified in step 1.

Step 3. Configuring the WebLogic Integration Database for the Domain

If you have not already done so, you must create the WebLogic Integration database tables for your domain. For detailed instructions, see "Configuring the Database for a Domain" in *Starting, Stopping and Customizing BEA WebLogic Integration*, which is available at the following URL:

```
http://edocs.bea.com/wlintegration/v2_1sp/config/index.htm
```

Step 4. Replacing the xmltoolkit.jar File

The BEA WebLogic Adapters include a new `xmltoolkit.jar` file. You must replace your existing WebLogic Integration `xmltoolkit.jar` file with the new JAR file.

To configure the new `xmltoolkit.jar` file, follow these steps:

1. Rename your original `xmltoolkit.jar` file to `xmltoolkit.jar.old` by entering the commands appropriate for your operating system:
   - On a Windows system:

     ```
     cd WLI_HOME\lib
     rename xmltoolkit.jar xmltoolkit.jar.old
     ```

   - On a UNIX system:

     ```
     cd WLI_HOME/lib
     mv xmltoolkit.jar xmltoolkit.jar.old
     ```

2. Extract the `xmltoolkit.jar` file from the adapter EAR file into a temporary directory.
3. Copy the new `xmltoolkit.jar` file (extracted in step 2) to the *`WLI_HOME`*`\lib` directory for Windows or the *`WLI_HOME`*`/lib` directory for UNIX.

Caution: Simply replacing the `xmltoolkit.jar` file is not sufficient; you must also make changes to the `setenv` and `Set`*`DomainType`*`Data` scripts as described in the following steps.

4. Edit the top-level `setenv` script and make the appropriate changes for your operating system:
   - On a Windows system, edit the *`WLI_HOME`*`\setenv.cmd` script.

     Replace the line:

     ```
     set WLICOMMONCP=
     ```

     with

     ```
     set WLICOMMONCP=%WLI_HOME%\lib\xmltoolkit.jar
     ```

- On a UNIX system, edit the *WLI_HOME*/setenv.sh script.

  Replace the line:

  ```
  WLICOMMONCP=$WLI_LIB/wlicommon.jar
  ```

  with

  ```
  WLICOMMONCP=$WLI_LIB/wlicommon.jar:$WLI_HOME/lib/xmltoolkit.
  jar
  ```

5. Edit the Set*DomainType*Data script.

   Here, *DomainType* is the type of the domain. For example, depending on the configuration of your domain, locate and edit the SetwliDomainData.cmd or SeteaiDomainData.cmd file.

   - On a Windows system:

     For example, edit the *DOMAIN_HOME*\SetwliDomainData.cmd script.

     Replace the line:

     ```
     set SVRCP=%WLISERVERCP%;%CMNCP%
     ```

     with

     ```
     set
     SVRCP=%WLI_HOME%\lib\xmltoolkit.jar;%WLISERVERCP%;%CMNCP%
     ```

   - On a UNIX system:

     For example, edit the *DOMAIN_HOME*/SetwliDomainData script.

     Replace the line:

     ```
     SVRCP=$WLISERVERCP:$CMNCP
     ```

     with

     ```
     SVRCP=$WLI_HOME/lib/xmltoolkit.jar:$WLISERVERCP:$CMNCP
     ```

Step 5. Updating the BEA License

The BEA WebLogic Adapter for SWIFT cannot be used without a valid software license. If you have downloaded the adapter for evaluation, you must obtain an evaluation license as described on the adapter download page. If you have purchased a license for the adapter, the license file is typically sent to you as an e-mail attachment.

When you have obtained a valid license for the adapter, update your `license.bea` file by completing the following steps:

1. Save the license file that you obtained with a name other than `license.bea`, in the *`BEA_HOME`* directory. For example, save the file as `swift_adapter_license.bea`. Use this file as the *`license_update_file`* in step 4 of this procedure.

   Warning: Do not overwrite or change the name of the existing `license.bea` file.

2. Perform the step appropriate for your platform:

   - On a Windows system, open an MS-DOS session and go to the *`BEA_HOME`* directory.
   - On a UNIX system, go to the *`BEA_HOME`* directory.

3. If it is not already included, add the JDK to your `PATH` variable by executing the command appropriate to your system:

   - On a Windows system:

     ```
     set PATH=BEA_HOME\jdk131_03\bin;%PATH%
     ```

   - On a UNIX system:

     ```
     PATH=BEA_HOME/jdk131_03/bin:$PATH
     export PATH
     ```

4. Merge the license update file into your existing license by executing the command appropriate to your system:
   - On a Windows system:

     `UpdateLicense` *`license_update_file`*
   - On a UNIX system:

     `sh UpdateLicense.sh` *`license_update_file`*

   Here, *`license_update_file`* is the name to which you saved the license update file in step 1.
5. Save a copy of your updated `license.bea` file in a safe place outside the WebLogic Integration and application installation directories.

Step 6. Deploying the Adapter Using the WebLogic Server Console

After the BEA WebLogic Adapter for SWIFT is installed, it must be deployed to WebLogic Server for your domain (for example, `wlidomain`). To configure and deploy an adapter using the WebLogic Server Administration Console, complete the following steps:

1. Start WebLogic Server.
2. Start the WebLogic Server Administration Console in a browser using the following URL:

   `http://`*`localhost:port`*`/console/`

   Here, *`localhost`* represents the machine on which WebLogic Server is running and *`port`* represents the listening port.

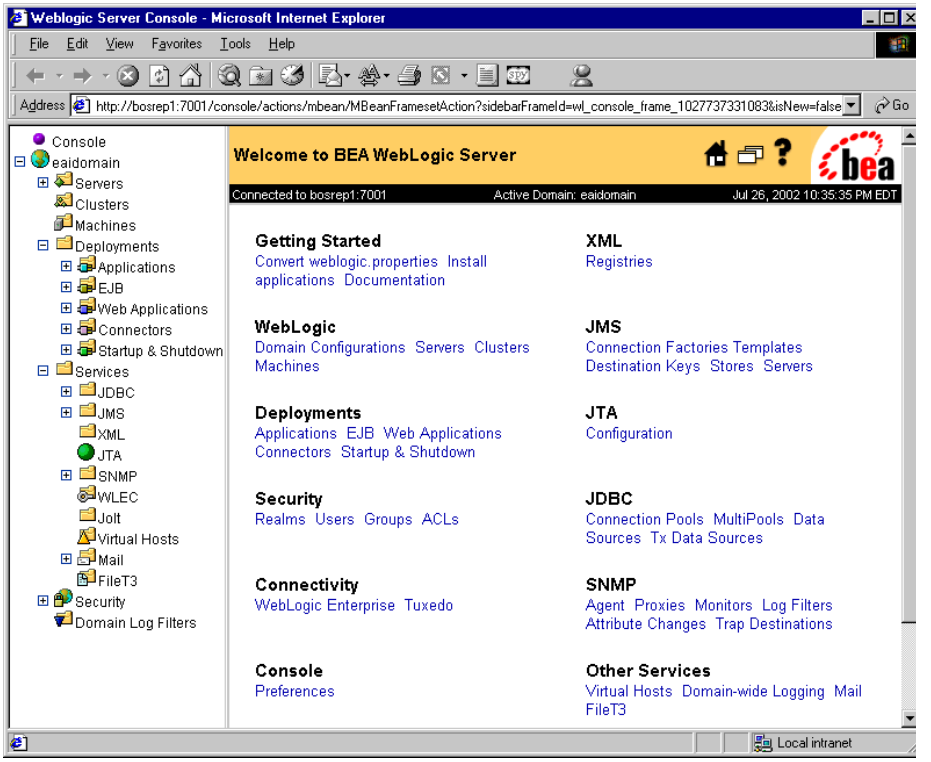
   For example, `http://localhost:7001/console/`
3. When prompted, enter the user name and password for the server.

Note: If you have not updated the default login, see "WebLogic Integration Users and Passwords" in *Starting, Stopping, and Customizing WebLogic Integration* at the following URL:

`http://edocs.bea.com/wlintegration/v2_1sp/config/getstart.htm`

The WebLogic Server Administration Console opens.

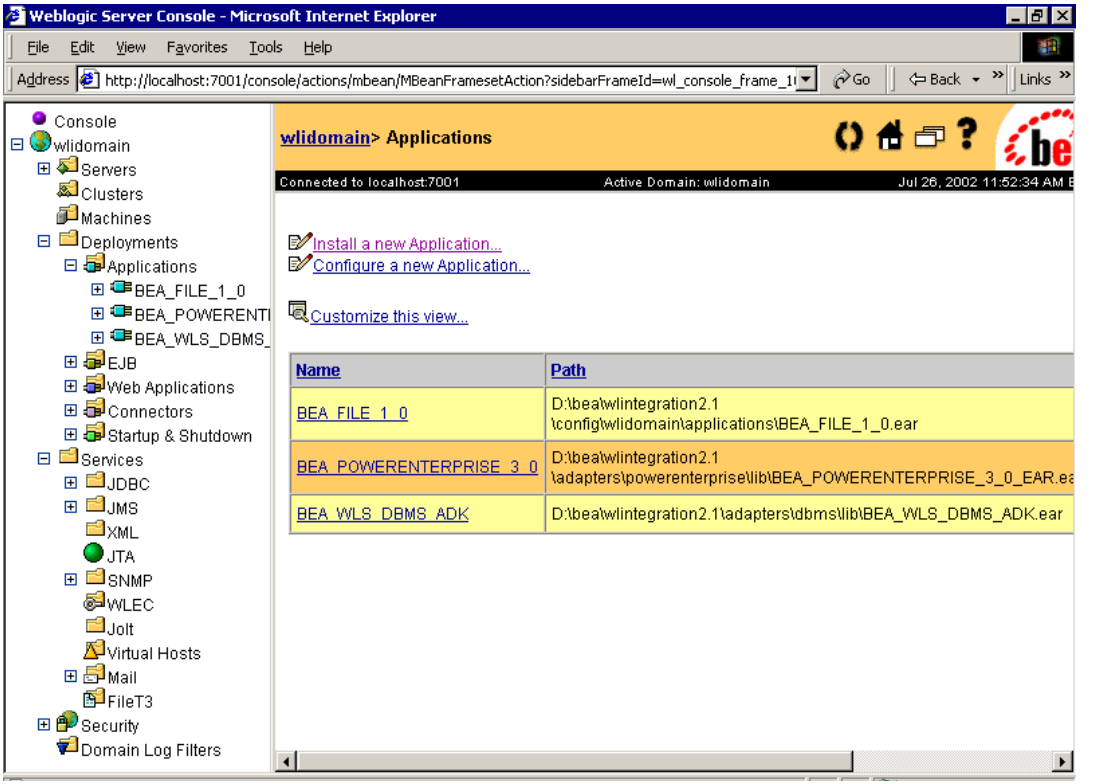
Figure 1-1 WebLogic Server Console



4. In the left pane, choose Deployments and then Applications from the navigation tree.

   The console displays the Applications window.

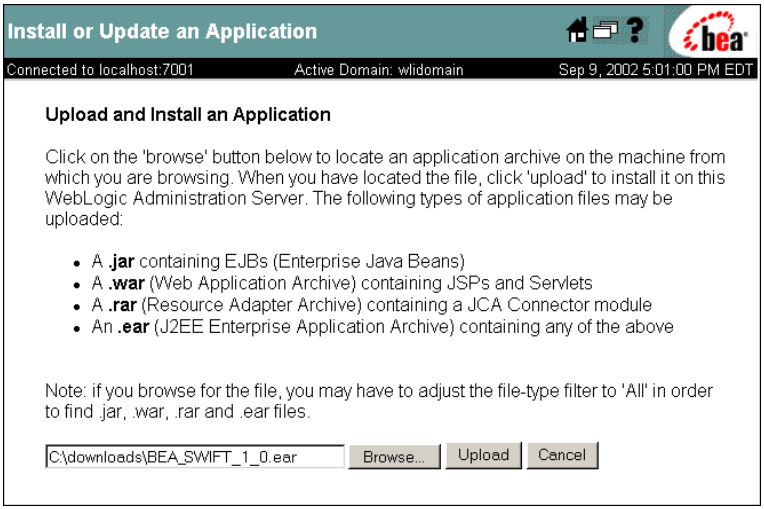
Figure 1-2 Applications Window



5. Click the Install a new Application link.

   The console displays the Install or Update an Application window.

Figure 1-3 Locating the Application Archive Window



6. Click Browse to locate the application archive you selected during installation (`BEA_SWIFT_1_0.ear`).
7. Click Upload to upload the `BEA_SWIFT_1_0.ear` file.

   The console displays the application files currently installed to indicate that the upload is complete and the adapter file is deployed to WebLogic Server.
8. You can verify deployment by viewing the adapter configuration, as follows:

   a. Choose Deployments and then Applications from the navigation tree.

   b. Click the `BEA_SWIFT_1_0.ear` file link.

Step 7. Adding the Administrative Server User Name to the Adapter Group

A user group, `adapter`, is defined in each domain that supports application integration functionality. Before you create an application view that employs the events or services supported by an adapter, you must add the user name defined for the administrative server to the `adapter` group.

Note: By default, the `adapter` group includes the user `system`. If the user name defined for the administrative server is `system`, skip this step. For example, if you are starting the server in a preconfigured domain, and you have not modified the default administrative server login, you can skip this step.

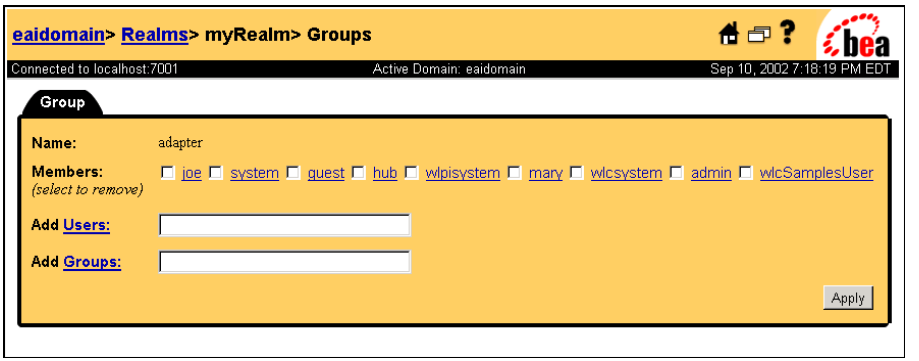
To add the administrative server user name to the `adapter` group, complete the following steps:

1. In the left pane of the WebLogic Server Administration Console, choose Security and then Groups from the navigation tree.

   The console displays groups currently defined for the domain.

2. Locate and click the link for the `adapter` group to display the group definition.

Figure 1-4 Group Definition



3. If the administrative server user name is not included in the Members list, enter the user name in the Add Users field.
4. Click Apply to add the user name to the group.

   The name is added to the Members list.

Before Using the BEA WebLogic Adapter for SWIFT with MQSeries as a Transport

To interact with IBM MQSeries (now known as WebSphere MQ), you must first install MQSeries or MQSeries Client on the machine where the BEA WebLogic Adapter for SWIFT will be installed. Additionally, you must install the MA88 SupportPac. This SupportPac provides support for developing MQSeries applications in Java (for deployment on MQSeries v5.2) through the following Java-based APIs:

- MQSeries classes for Java
- MQSeries classes for Java Message Service (JMS)

The SupportPac may be downloaded from the following URL:

`http://www-3.ibm.com/software/ts/mqseries/txppacs/ma88.html`

and is available on the following platforms:

- AIX
- HP-UX 10-20
- HP-UX 11
- iSeries
- Linux for Intel
- Linux for S/390
- Microsoft Windows 95
- Microsoft Windows 98

- Microsoft Windows NT
- Microsoft Windows 2000
- OS/390 V2R9 or higher (including z/OS)
- Sun Solaris

Next Steps

When you have successfully installed and deployed the adapter, you can log on to the WebLogic Integration Application View Console to create application views that employ events and services supported by the BEA WebLogic Adapter for SWIFT. For more information, see the *BEA WebLogic Adapter for SWIFT User Guide*.